UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/623,101	07/18/2003	Guillermo Rozas	TRAN-P072	2896

7590 09/17/2007

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
PETRANEK, JACOB ANDREW	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
09/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/623,101	ROZAS ET AL.	
	Examiner	Art Unit	
	Jacob Petranek	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 August 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-14 are pending.

2. The office acknowledges the following papers:

Claims and arguments filed on 8/14/2007.

New Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-3, 5-7, 9-12, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Larson (U.S. 5,115,500).

5. As per claim 1:

Larson disclosed a method of processing an instruction, said method comprising:

Fetching said instruction using a corresponding address from a memory unit (Larson: Figure 2, column 5 lines 34-67 continued to column 6 lines 1-40)(An instruction is fetched from the I-Store 2 using an address from the memory unit (Instruction Address Register, IAR 3).);

Wherein a plurality of possible meanings are associated with said instruction by the same processor (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12 and column 5 lines 34-67 continued to column 6 lines 1-40)(There are a possibility of a plurality of meanings for each instruction depending on the concatenated address

bits. When an instruction is fetched, it has one of two meanings, which are an instruction of type 1 if the lower three order bits aren't '111' and an instruction of type 2 if the lower three order bits are '111.' The objective of the invention is to allow for execution of two or more different machine types on a single processor. Thus, it's obvious to one of ordinary skill in the art that the type 1 and type 2 instructions are executed by the same processor.);

Concatenating a portion of said corresponding address to said instruction to form an extended instruction, wherein selection of said portion of said corresponding address for said concatenating is independent of region of said memory unit from which said instruction is fetched (Larson: Figure 2, column 3 lines 52-64 and column 5 lines 34-67 continued to column 6 lines 1-40)(The selection of the three lower end bits is independent of which region of memory the instruction is fetched from. The three bits are selected to determine how the instruction will be decoded regardless of if the instruction came from the memory units that store type 1 instructions or the memory units that store type 2 instructions.),

And wherein said concatenation increases a number of instructions in an instruction set (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12)(The instruction set architecture is comprised of both the type 1 and type 2 instruction sets. Prior to the concatenation, only a single type of instructions was allowed to execute. However, now with the concatenation, the number of instructions can at the maximum double the number of opcodes available for instructions. Type 1

and type 2 are different instructions, which results in increasing the number of instructions available to the overall instruction set architecture of the processor.); and

Executing said extended instruction, wherein said portion of said corresponding address determines a meaning for said extended instruction from said possible meanings (Larson: Figure 2, column 5 lines 34-67 continued to column 6 lines 1-40)(The instruction is decoded and then executed with one of the possible meanings, which is dependent on the extended instruction formed from the concatenation.).

6. As per claim 2:

Larson disclosed the method as recited in claim 1 wherein said portion is an address bit (Larson: Fig. 2, col. 5, line 34 to col. 6, line 40).

7. As per claim 3:

Larson disclosed the method as recited in claim 1 wherein said portion is a plurality of address bits (Larson: Fig. 2, col. 5, line 34 to col. 6, line 40).

8. As per claim 5:

Larson disclosed a method of handling an instruction, said method comprising:

Generating said instruction, wherein a plurality of possible meanings are associated with said instruction by a same processor (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12 and column 5 lines 34-67 continued to column 6 lines 1-40)(There are a possibility of a plurality of meanings for each instruction depending on the concatenated address bits. When an instruction is fetched, it has one of two meanings, which are an instruction of type 1 if the lower three order bits aren't '111' and an instruction of type 2 if the lower three order bits are '111.' The objective of

the invention is to allow for execution of two or more different machine types on a single processor. Thus, it's obvious to one of ordinary skill in the art that the type 1 and type 2 instructions are executed by the same processor.);

Storing said instruction at a particular address in a memory unit such that a portion of said particular address enables determination of a meaning for said instruction from said possible meanings (Larson: Figure 2, column 2 lines 21-54 and column 5 lines 34-67 continued to column 6 lines 1-40.);

And before executing said instruction, fetching said instruction using said particular address from am memory unit and concatenating said portion of said particular address to said instruction, wherein selection of said portion of said corresponding address for said concatenating is independent of region of said memory unit from which said instruction is fetched (Larson: Fig. 2, col. 3, lines 52-64 and col. 5, line 34 to col. 6, line 40)(The selection of the three lower end bits is independent of which region of memory the instruction is fetched from. The three bits are selected to determine how the instruction will be decoded regardless of if the instruction came from the memory units that store type 1 instructions or the memory units that store type 2 instructions.),

And wherein said concatenation increases a number of instructions in an instruction set (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12)(The instruction set architecture is comprised of both the type 1 and type 2 instruction sets. Prior to the concatenation, only a single type of instructions was allowed to execute. However, now with the concatenation, the number of instructions

can at the maximum double the number of opcodes available for instructions. Type 1 and type 2 are different instructions, which results in increasing the number of instructions available to the overall instruction set architecture of the processor.).

9. As per claim 6:

Claim 6 essentially recites the same limitations of claim 2. Therefore, claim 6 is rejected for the same reasons as claim 2.

10. As per claim 7:

Claim 7 essentially recites the same limitations of claim 3. Therefore, claim 7 is rejected for the same reasons as claim 3.

11. As per claim 9:

Larson disclosed the method of as recited in claim 5 wherein said generating said instruction and said storing said instruction are performed by a compiler (Larson: Column 1 lines 11-29)(Larson disclosed that a compiler must be generated for each new machine. A compiler by definition translates high-level language into object code prior to the execution of a program. Thus, a compiler generates instructions that are executable on a processor. A compiler is also defined as any program that transforms one set of symbols into another by following a set of syntactic and semantic rules. As shown in figure 2, a rule for the processor is that type 2 instructions can only be placed in memory locations ending with '111.' Thus, it's obvious to one of ordinary skill in the art that this is a semantic rule that the compiler of Larson must follow and correctly place all type 2 instructions only in memory locations ending with '111' and place all

type 1 instructions at other memory locations. Thus, the compiler also stores instructions in memory places.).

12. As per claim 10:

Claim 10 essentially recites the same limitations of claim 1. Therefore, claim 10 is rejected for the same reasons as claim 1.

13. As per claim 11:

Claim 11 essentially recites the same limitations of claim 2. Therefore, claim 11 is rejected for the same reasons as claim 2.

14. As per claim 12:

Claim 12 essentially recites the same limitations of claim 3. Therefore, claim 12 is rejected for the same reasons as claim 3.

15. As per claim 14:

Claim 14 essentially recites the same limitations of claim 9. Therefore, claim 14 is rejected for the same reasons as claim 9.

16. Claims 4, 8 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Larson (U.S. 5,115,500), in view of ("390 Principles of Operation"), herein referred to as IBM.

17. As per claim 4:

Larson disclosed the method as recited in claim 1.

Larson failed to teach wherein the plurality of possible meanings include an integer type of instruction and a floating point type of instruction.

However, IBM disclosed wherein the plurality of possible meanings include an integer type of instruction and a floating point type of instruction (IBM: Pages 7-1 to 7-6, 9-1 to 9-4, and 9-8 to 9-9)(The combination results in type 1 instructions being the instructions of the IBM 390 ISA. Thus, one of the plurality of possible meanings could be a integer instruction from the 390 ISA or a floating point instruction from the 390 ISA.).

Larson disclosed two separate types of instruction used, but failed to disclose what types of ISA's are used. Since the Larson patent was produced from IBM, one of ordinary skill in the art would have been motivated to look at IBM ISA's for more information on what types of instructions are supported. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the IBM 390 ISA onto the processor of Larson.

18. As per claim 8:

Claim 8 essentially recites the same limitations of claim 4. Therefore, claim 8 is rejected for the same reasons as claim 4.

19. As per claim 13:

Claim 13 essentially recites the same limitations of claim 4. Therefore, claim 13 is rejected for the same reasons as claim 4.

Response to Arguments

20. The arguments presented by Applicant in the response, received on 8/14/2007 are not considered persuasive.

21. Applicant argues "Larson discloses that the same instruction is decoded differently such that it can be executed on different machines based on where the instruction is stored; however, each decode nevertheless leads to the same meaning, e.g., an add instruction. Thus, the number of instructions remains unchanged ..."

This argument is not found to be persuasive for the following reason. The invention combined two different instruction sets into a single instruction set onto the same processor. It doesn't matter if the type 1 and type 2 share the same opcode for a particular instruction because they are still two individual instructions within the overall instruction set architecture. This can be shown in figure 3 that shows the instructions of type 1 and type 2 instructions are of different bit sizes. Therefore, the instructions for each the type 1 and type 2 contribute to the total number of instructions for the overall instruction set architecture of the processor.

22. Applicant argues "Larson fails to either teach or suggest an instruction having different meanings associated with it by the same processor, as claimed."

This argument is not found to be persuasive for the following reason. Larson discloses in column 4 lines 56-67 continued to column 5 lines 1-12 that the objective of the invention is to allow for execution of two or more different machine types on a single processor. Thus, it's obvious to one of ordinary skill in the art that the same processor executes the type 1 and type 2 instructions.

23. Applicant argues "Larson doesn't teach or suggest wherein a plurality of possible meanings are associated with said instruction."

This argument is not found to be persuasive for the following reason. There is a plurality of meanings for each instruction depending on the concatenated address bits. When an instruction is fetched, it has one of two meanings, which are an instruction of type 1 if the lower three order bits aren't '111' and an instruction of type 2 if the lower three order bits are '111.' In addition, the claimed limitation doesn't limit the plurality of possibilities to be limited to a single ISA.

24. Applicant argues "Larson doesn't teach or suggest concatenating a portion of said corresponding address to said instruction to form an extended instruction, wherein selection of said portion of said corresponding address for said concatenating is independent of region of said memory unit from which said instruction is fetched."

This argument is not found to be persuasive for the following reason. The selection of the three lower end bits is independent of which region of memory the instruction is fetched from, as these three bits are always selected. The three bits are selected to determine how the instruction will be decoded regardless of if the instruction came from the memory units that store type 1 instructions or the memory units that store type 2 instructions.

25. Applicant argues "IBM fails to disclose that the plurality of possible meanings include an integer type and a floating point type."

This argument is not found to be persuasive for the following reason. The examiner brought in IBM to show an instruction set that includes both integer and floating point operations and stated that the IBM ISA would be used as type 1 instructions on the processor. Both integer and floating-point instructions are very

common for instruction set architectures. Thus, it's obvious to one of ordinary skill in the art that both the type 1 and type 2 instruction sets can include integer and floating-point types of instructions. It's obvious to one of ordinary skill in the art that a compiler could place a floating-point instruction into a '111' memory slot for type 2 and that the instruction could otherwise be interpreted as a integer instruction in a type 1 instruction. This is obvious to one of ordinary skill in the art because the type of instruction to be executed is solely dependent on the opcode of the instruction. One of ordinary skill in the art also is very aware of the fact that opcodes can be arbitrarily assigned to instructions during the initial design phase of a processor when selecting instructions to be included within the instruction set architecture. Thus, it's obvious to one of ordinary skill in the art that an instruction from a first instruction set could be of an integer type and that when the instruction is interpreted in view of the other instruction set architecture, the opcode would define the instruction as a floating-point operation.

26. Applicant challenges the official notice given to claims 9 and 14 and requests the examiner provide documentation for the rejections.

The examiner has given claims 9 and 14 a new ground of rejection and withdrawn the previous official notice.

Conclusion

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the

claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jacob Petranek
Examiner, Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100